Assessing Student Learning Across Various Database Query Languages

Zepei Li, Sophia Yang, Kathryn Cunningham, and Abdussalam Alawini
Department of Computer Science
University of Illinois Urbana-Champaign
Urbana, IL, United States of America
{zepeili2, sophiay2, katcun, alawini}@illinois.edu

2023 IEEE Frontiers in Education Conference (FIE) | 979-8-3503-3642-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/FIE58773.2023.10343409

***Abstract*—Previous research has shown that students encounter difficulties when learning database systems and their corresponding languages. Researchers have categorized these challenges into syntax and semantic errors and have identified common error types and overall learning obstacles among students. However, most existing studies have primarily focused on quantitatively assessing students' overall performance in an aggregated manner, which may overlook valuable insights into individual-level knowledge transfer. In this study, we scrutinized over 250,000 submissions to query language programming assignments, their corresponding error messages, and the performance data of 702 students who took a database course in the Fall 2022 semester at the University of Illinois Urbana-Champaign to gain a comprehensive overview of each student's performance. We followed each student's progress in semantic and syntax errors across three query languages to determine their overall learning experience and whether knowledge transfer had occurred. Consequently, we discovered that many students may still encounter difficulties when transferring their knowledge from one language to another, despite having already learned and practiced the same abstract data operation concepts in one language. On the other hand, the majority of students were able to reduce syntax errors through practice in one language, but the rate of improvement varied among individuals. This study seeks to investigate two key aspects: the potential transfer of abstract data operation concepts among different database languages, and the possibility of a decrease in syntax errors through consistent practice within a single query language.**

***Index Terms*—Structured Query Language (SQL), MongoDB, Neo4j, semantic errors, syntax errors, knowledge transfer**

I. Introduction

As digital technology progresses, organizations that utilize data-driven decision-making are becoming more aware of the significance of data collection, storage, organization, and query processes. Consequently, a firm grasp of query language skills has become increasingly necessary for users, developers, and professionals [1]. Database education is therefore indispensable, as it provides individuals with the skills and knowledge necessary to design, develop, and maintain efficient and effective databases. Due to the existence of various database models (types), such as relational, graph, or document-oriented, each corresponds with its distinct query languages and data modeling features; it has become vital to possess a versatile skill set in database query languages. To address this demand, many educational institutions have adopted a curriculum that combines relational and NoSQL ("not only SQL") databases [2]–[5]; Yet, database courses typically teach each data model separately without providing insights into their trade-offs, which may limit students' ability to generalize their knowledge and transfer them to new domains or database types [6].

For instance, Structured Query Language (SQL) has an English-like syntax that makes it more accessible to novices with no prior programming experience, indicating that it could be a potential gateway into computing. Meanwhile, NoSQL databases and their query language, such as MongoDB, are better suited for distributed data stores with flexible data types. In comparison, graph databases like Neo4j excel in representing highly interconnected data and offer efficient retrieval and navigation capabilities. Consequently, database instructors may face the challenge of determining whether to teach multiple query languages, if students exhibit effective knowledge transfer, and the sequence of query languages to teach. Although the acquisition of database query languages is critical to one's skill set and many institutions offer database courses, we know very little about how to support students as they are learning multiple query languages, and there is a relative dearth of research in this area to investigate these questions. Therefore, our study addresses two research questions.

RQ1 *Do students transfer knowledge of abstract data operations across database languages?*

RQ2 *As students practice the same query language, do they reduce their syntax errors?*

The first research question explores learning transfer among three query languages (SQL, MongoDB, and Neo4j) in the context of basic abstract data operations (discussed in Section III). The objective of this investigation is to analyze the occurrence of semantic errors when learners attempt to apply these abstract operations across different query languages. The second question investigates the impact of practicing the same query language, particularly MySQL, on the reduction of syntax errors. By emphasizing syntax in this part, we aim to assess the transfer of syntactical knowledge through practice in homework assignments within a single language. This two-part approach allows us to gain insights into both the challenges and opportunities of learning transfer in the domain of databases, contributing to a comprehensive understanding of the transferability of concepts across query languages and the role of practice in knowledge transfer within the same query language.

II. Literature Review

In our study, syntax errors are defined as errors that occur during query compilation, resulting in the query's failure to execute, whereas semantic errors occur when a query produces incorrect or unintended results due to errors in the query's logic or structure. Prior research work has examined the common syntax and semantic errors that students encounter in SQL [7]–[12]. However, no research has been designed to examine students on an individual level, following their progression in learning over a period of time, and especially across different query languages (to evaluate the transfer of concepts). Furthermore, existing research on programming languages also discovered and assumes that syntax errors will be reduced through practice [13]; however, there has been no evidence directly linked to the reduction in syntactic errors through query formulation practice.

SQL is the primary query language for relational databases and has become the industry standard for data retrieval and manipulation [14], [15]. Due to its English-like syntax, it is an accessible gateway into computing, particularly for novices. Additionally, being a standardized language, SQL is often regarded as a universal language for database management [16]. As a result, within the realm of SQL education, researchers have focused on several areas to help students better understand and master this important language, including syntax errors [9], [10], semantic errors [8], [10], and logic errors [10] that students often find challenging. Researchers have investigated the SQL query concepts that lead to persistent error types [11], the underlying reasons why students encounter these errors [17], what makes query formulation difficult [12], [18], [19], categorizations of semantic errors [20], and visualizations that help students to understand SQL queries [21] or instructors to understand student's learning progressions and challenges [22]–[24].

MongoDB is a NoSQL database system that stores data in flexible schema documents, enabling developers to modify the data model quickly without pre-defining or redesigning the entire database schema [25]. As the popularity of MongoDB continues to increase among users [26], researchers have emphasized the importance of integrating NoSQL databases into the curriculum [2]–[5], [27]. Additionally, some researchers have analyzed the errors that students made on homework assignments in MongoDB and have developed a categorization for them [28].

Neo4j is a widely used graph database management system that employs Cypher, a flexible query language with declarative pattern-matching features and a syntax similar to SQL, tailored for graph data analysis [6], [29]. This combination offers a powerful and efficient solution for managing complex, interrelated data structures [29], [30]. With these advantages in mind, researchers in the domain of graph database education have conducted a quantitative analysis of student errors on Neo4j coding assignments and identified challenging concepts that students encountered [31].

Although previous studies have focused on the challenges encountered by students in various query languages, they have not adequately investigated the aspect of knowledge transfer (i.e. *the application of knowledge, skills, or information from one context to another, enhancing problem-solving abilities and improving overall understanding of the subject matter* [32], [33]) across different query languages. In addition, not all transfer is the same. Researchers further categorize knowledge transfer into more detailed divisions, such as near transfer (knowledge transfer between more similar contexts) and far transfer (between contexts that differ in more ways) [34]. The use of abstract data operations across languages and syntactic knowledge within one language are both transferable types of knowledge because their underlying concepts remain constant, even though their application methods may differ across various scenarios [6]. However, the application of syntax structures within the same language is nearer transfer than the application of abstract data operations across languages.

III. Background

We first start by defining data models to establish the conceptual framework for comparing different query languages. In data management, a data model serves as an abstract framework that arranges data elements, establishing relationships between them and aligning them with the attributes of real-world entities. It comprises three essential components: Structure, Operation, and Constraint [35]. The *Structure* component refers to how data is organized and represented within the model. For example, in SQL, the structure of a relational database is based on relations (tables) where tuples (rows) represent records and columns represent attributes. The *Operation* component defines how data can be manipulated and accessed within the model. In MongoDB, operations such as *find()* (finds documents that match criteria), *aggregate()* (using pipeline stages to process data records and return computed results), and *insertOne()* (inserts a single new document into a collection) enable data manipulation and retrieval in a document-based model. The *Constraint* component encompasses the rules or conditions that the data must comply with. Some of the most common types of constraints in Neo4j include *Unique Constraints* and *Node Key Constraints* which enforce certain rules and conditions on nodes and relationships to ensure that the graph data model adheres to the intended design and business rules. These components collectively determine how data is organized, manipulated, and controlled.

Although SQL, MongoDB, and Neo4j possess distinct data structures and constraints, they share common abstract data operations such as *Selection*, *Projection*, and *Aggregation* [6]. Selection involves filtering rows or records of data from a larger dataset based on specified conditions or criteria. Projection focuses on obtaining specific columns or attributes from a dataset while excluding others. Aggregation performs calculations on a set of values to derive a single value. These fundamental operations enable data manipulation and analysis, allowing users to extract, filter, organize, and summarize data according to specific requirements. Our study builds upon this existing classification of operations to examine whether students can transfer knowledge across different query languages or if they encounter challenges with the same abstract operations across various database query language homework assignments.

A. Homework Overview

In this section, we will provide an overview of the homework assignments and the concepts they examine. Below is an example of a sample SQL homework problem and its corresponding instructor solution:

> *Write one SQL query that returns for each student (NetId), the total number of 3 and 4 credit courses (as course_count) they have taken and their average grade (as avg_grade) in these courses. Please round the average grade to the second decimal place. You can do so by applying a ROUND(x, 2) function where x is the original value. Return the results in ascending order of the NetId.*

```
SELECT S.netId, COUNT(*) as course_count,
    ROUND(AVG(Score),2) as avg_grade
FROM Students S JOIN Enrollments E ON S.NetId
    = E.NetId
WHERE E.Credits = 4 OR E.Credits = 3
GROUP BY E.NetId
ORDER BY S.netId
```

In SQL, the keyword *SELECT* is used to perform the Projection operation, allowing users to specify the columns or attributes they want to retrieve from a dataset. The keyword *WHERE* is used for Selection, enabling users to filter and retrieve specific rows based on specified conditions. The keywords *GROUP BY*, *AVG*, and *COUNT* are used for Aggregation, allowing users to group data based on specific attributes and perform calculations. For the other two languages, namely MongoDB and Neo4j, the abstract data operations they perform are similar to SQL, although the syntactic structures and keywords differ.

The SQL homework consists of 15 questions, and the concepts examined for each question (represented by Q1 to Q15) are presented in Table I.

Apart from SQL, the MongoDB component of the course includes 12 homework problems that utilize MongoDB's JavaScript shell as the query language while Neo4j consists of 8 questions that use Cypher as the query language.

IV. Methodology

Our study follows a two-part approach to investigate the challenges and opportunities of learning transfer of concepts across languages and the role of practice in knowledge transfer within the same query language.

In the first part, we explore the learning transfer among three query languages in the context of abstract data operations. Our goal is to analyze the occurrence of semantic errors when learners attempt to apply these abstract operations across different query languages. In the second part, we investigate the impact of practicing the same query language, particularly MySQL, on the reduction of syntax errors. We aim to assess the transfer of syntactical knowledge through practice in homework assignments within a single language.

TABLE I: SQL Concepts per Question

Exercise	Concepts
Q1	Single-table queries with complex where clauses I
Q2	Single-table queries with complex where clauses II
Q3	Join and where
Q4	Join and correlated subqueries
Q5	Complex Groupby and subqueries
Q6	Basic Groupby and Aggregation
Q7	Update
Q8	Complex Groupby and Having
Q9	Complex Groupby, Having and subqueries
Q10	Set operations (UNION)
Q11	Complex Join, Aggregation, subqueries
Q12	Outer Join, Groupby with having, subqueries
Q13	Complex Joins, Groupby, subqueries
Q14	Trigger
Q15	Stored procedure

Our hypothesis is that students with a high number of errors typically have a weaker understanding of the topic than students with a relatively low number of errors. Furthermore, if knowledge transfer occurs successfully through learning and continuous practice, we should observe a downward trend in the number of student errors in questions with similar concepts. Conversely, we may observe a fluctuation or increase in the number of errors if the knowledge transfer is incomplete or unsuccessful. We analyzed the student's homework submissions to validate our research questions and hypothesis.

A. Data Handling

We analyzed the homework submissions for query languages related to the relational, document-oriented, and graph data models from the Fall 2022 semester database course at the University of Illinois Urbana-Champaign. These submissions were acquired through PrairieLearn [36], an online assessment and learning system. The query files were anonymized by removing any identifiable information, and a random number was assigned to each student as their unique identifier to build connections between their performance in different languages.

B. Learning Transfer Across Different Languages

We chose to focus on semantics in this part to better understand the core comprehension and conceptual difficulties associated with learning transfer across different languages while avoiding confounding factors such as syntax variations that could obscure the analysis of learning transfer among the syntactically different query languages. To achieve this, we identified the concepts tested in each homework assignment and created a concept map to connect these concepts to the corresponding abstract data operations in each database query language. Mapping assignment questions to abstract data operations allowed us to identify questions that test the same abstract data operation concepts across the three query language assignments. We then grouped those questions based on the concepts they tested to better understand students' performance among those question groups. Two groups (S1 and S2) are appropriate for comparison, and they are listed below:

- S1 contains Question (Q)1 in SQL, Q1 in MongoDB, and Q1 in Neo4j. All questions examine the concept of using basic Selection and Projection.
- S2 contains Q6 in SQL, Q2 in MongoDB, and Q3 in Neo4j. These questions mainly examine the concepts of Aggregation.

For S1, all questions within the group require students to extract out specific data sets from the database based on given criteria. For S2, the concepts being tested include Selection and Projection, as well as Aggregation, which is a new concept introduced in the homework assignments.

The database course for this semester is divided into two sections, A and Q. Section A teaches the database languages in the order of MongoDB, SQL, and Neo4j, while Section Q teaches SQL, MongoDB, and then Neo4j. Both sections use identical materials, follow the same teaching design, and are taught by the same instructor. Based on groups, we analyzed the number of semantic errors made by individual students in Sections A and Q.

C. *Learning Transfer Within a Single Language*

In this part of our study, we shifted our focus to syntax which serves as a finer-grained indicator (as opposed to semantics) for evaluating the effectiveness of practice in reducing syntactical errors and improving learners' proficiency within the same language. We analyze the number of syntax errors to test if syntactical knowledge can be transferred through practice in homework assignments, as reflected by the reduction of syntax errors. Due to space limitations, we focus on the most representative and comprehensive assignment, SQL, which has unique advantages for the broadest range of abstract operations coverage and the largest number of students who finished the assignments. We believe that syntactic analysis of SQL assignments can provide a more comprehensive insight into knowledge transfer in database query languages.

In SQL, different concepts are associated with different SQL keywords, which have unique syntactic structures. Therefore, we also grouped the assignments according to the concepts examined in the questions. We selected three groups (G1, G2, G3) of questions suitable for comparison and showcase them below:

- G1 includes Q1 and Q2. Both questions examine the concept of using basic Selection and Projection.
- G2 includes Q5 and Q9. Both questions examine the concepts of Aggregation, Join, and Subqueries.
- G3 includes Q11 and Q13. Both questions dive into the same concept explored in the second group of questions. These advanced questions involve more complex requirements and involve the joining of multiple tables.

It is noteworthy that the SQL homework questions are structured with an ascending level of difficulty and complexity, indicated by increasing question numbers (Q1 to Q15). For example, Q2 involves more complex selection criteria compared to Q1. In addition, we find that the majority of students completed the first question (Qfirst) before the second (Qsec) in each of the question groups. Thus, for all the question groups considered, we calculated the error difference, defined as: Error Difference $= \text{Qfirst}_{SyntaxErrors} - \text{Qsec}_{SyntaxErrors}$ to examine students' progress. A positive Error Difference indicates a reduction in syntax errors, while a negative value suggests an increase in syntax errors between the two questions being compared.

V. Results

In this part, we will present our findings in the order of the research questions and provide explanations for the data. We will further discuss and analyze these results in Section VI.

A. *Semantic Error Progression*

The two sections with different instruction sequences behaved differently in terms of semantics. In general, a higher percentage of students in Section A showed a decreasing trend in semantic errors compared to Section Q. However, even within Section A, only around one-third of the students consistently demonstrated a reduction in semantic errors.

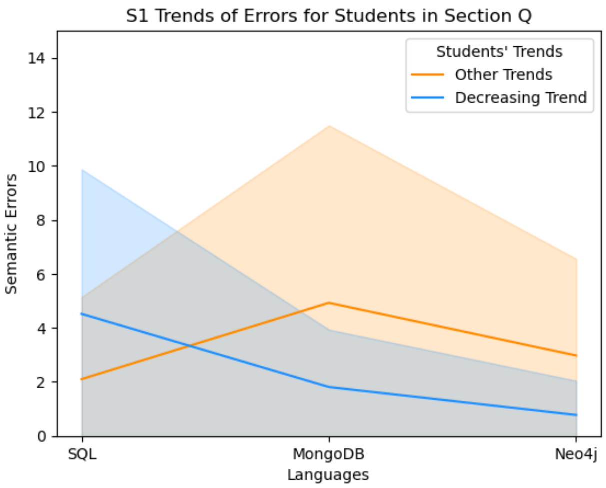

Fig. 1: The x-axis of the plot represents three specific questions in each language for the semantic S1 analysis. On the y-axis, we have the number of semantic error counts. The lines in the plot represent the average semantic count for different trends, while the shaded area represents the standard deviation band around the line.

In S1, which focuses on basic Selection and Projection across three languages, Figure 1 shows two major trends that students in Section Q exhibited. The blue part of the graph represents an aggregation of all the students who demonstrated a consistent reduction in errors across the three languages. In contrast, the orange part of the graph represents students who did not show a consistent decrease in errors. On average, students made 2.63 errors in SQL, 4.13 in MongoDB, and 2.42 in Neo4j. We refer to Figure 2 to provide a more detailed view. As depicted in the diagram, a considerable portion of the students (193 out of 372) experienced an increase in semantic errors in their second language, MongoDB. In the subsequent language, Neo4j, the majority of students (248 out of 372) displayed a recovery trend by successfully reducing their errors. However, overall, only 23.9% (89 out of 372) of students consistently reduced semantic errors following the learning path. And there was still a subset of students who consistently exhibited an increase in errors.

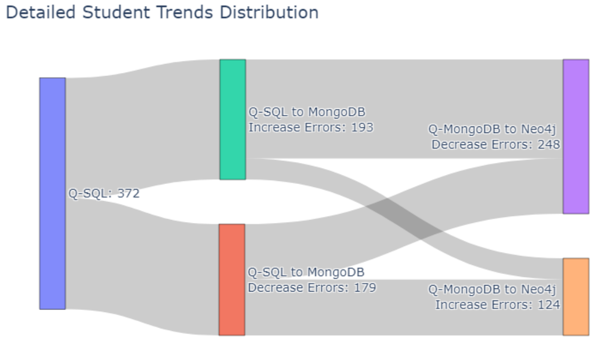

Fig. 2: The diagram illustrates the overall error changes among the students in Section Q. Each pipeline represents a stage of learning and is connected to the previous language. Thicker lines indicate a higher number of students exhibiting that particular trend, while thinner lines represent fewer students with that trend. A *Decrease* label in the graph indicates a reduction in semantic errors compared to the previous language and vice versa for an *Increase* label.

In Section A, as shown in Figure 3, students exhibited distinct trends, particularly among those who did not consistently reduce their errors (the orange part) when compared to students in Section Q. On average, we observed a relatively flat trend for the increment of errors in Section A. While it is true that a higher percentage of students in Section A demonstrated a consistent reduction in errors compared to Section Q, it is important to note that this improvement was not observed across all students. Specifically, only 33.9% of students in Section A (76 out of 224) consistently showed a reduction in semantic errors (color in blue) throughout their learning journey.

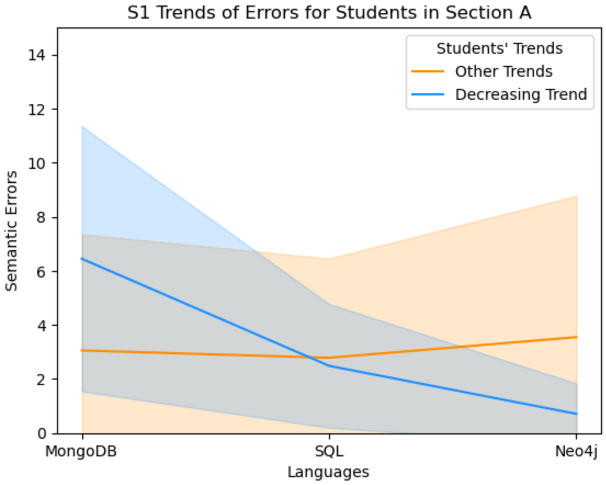

Fig. 3: Similarly to Figure 1, the x-axis of the figure represents the questions in each language for the semantic S1 analysis. The y-axis represents the number of semantic error counts. The lines on the plot depict the average semantic count for different trends, and the shaded area represents the standard deviation band around the line. The difference between Figures 1 and 3 is the sequence in which the languages were taught.

For S2, which primarily examines the new concept of Aggregation, the average number of semantic errors is presented in Table II. Due to the increased complexity and number of concepts examined in S2, it is expected that the average number of errors per question has increased compared to S1. Despite these differences, we observe a similar trend in errors, with the majority of students still making the most semantic errors in the MongoDB questions. In Section Q, only 47 out of 375 students (12.53%) were able to successfully reduce their errors in all three languages. In Section A, 78 out of 224 students (34.82%) demonstrated a continuous reduction in semantic errors. We discovered that there was a limited number of students who consistently reduced their errors across both sections and question groups. We will go into the potential reasons behind this observation in Section VI.

TABLE II: Average Errors in Different Sections for S2

Section	**SQL Errors**	**MongoDB Errors**	**Neo4j Errors**
Q	2.79	6.85	3.52
Section	**MongoDB Errors**	**SQL Errors**	**Neo4j Errors**
A	7.42	2.63	3.19

B. *Syntax Error Progression*

Our observations indicate that, in contrast to semantic errors, the majority of students in both sections displayed a trend of reduction in syntax errors in all three SQL homework groups. However, students with varying performances (as determined by the total number of submissions in each question group) exhibited diverse progressions. Table III presents the difference in the percentage of individuals exhibiting reduced syntax errors between the top 50% (High Total Submission) and bottom 50% (Low Total Submission) of the total submissions across various question groups.

For G1, in Section Q, a total of 348 out of 448 students (77.68%) displayed progress. Despite the majority of students neither increasing nor decreasing their syntax errors as Figure 4 shows, it is worth noting that the answer for Q2 was slightly longer than Q1. Therefore, if students maintained the same number of syntax errors, it can still be seen as

TABLE III: Group Performance Comparison

Question Group	Low Total Submission	High Total Submission
G1	189 / 224 (84.38%)	159 / 224 (70.98%)
G2	123 / 189 (65.08%)	144 / 189 (76.19%)
G3	154 / 210 (73.33%)	156 / 210 (74.29%)

a sign of progress. Additionally, we observed that as the number of total submissions increased, students displayed greater variations in their Error Difference compared to those with fewer submissions.

As we are comparing mastery of syntax knowledge within one language, we used the data from Section A to verify the findings from Section Q. In Section A, we saw a group of high-submission students who showed a reluctance to reduce errors which is similar to Section Q. Although students who did not display a reduction in syntax errors distributed differently, the percentage of students who showed progress remained relatively consistent. In Section A, 183 out of 255 students (71.48%) reduced their syntax errors. Due to the limitation of space, we will only report numerical statistics for Sections A and Q in the remaining question groups.

For G2, 267 out of 378 students (70.63%) and 175 out of 256 students (68.75%) showed a trend of syntax error reduction for sections Q and A, respectively. Although the percentage of students who reduced errors remained relatively consistent, the distribution of these students varied. Unlike in G1, where students with a higher number of total submissions exhibited more extreme variations, in G2 the students who reduced errors were more evenly distributed across different levels of total submissions. Furthermore, as shown in Table IV, the average number of reduced syntax errors increased significantly from G1 to G2.

In G3, a similar trend of reduction in syntax errors was observed. Among Section Q students, 310 out of 420 students (73.81%) showed a decreasing trend, while in Section A, 198 out of 256 students (77.34%) exhibited a reduction in syntax errors. The distribution of students who did not demonstrate a reduction trend was similar to G2. The average Error Difference remained at a similar level as G2. These findings will be further discussed in Section VI.

TABLE IV: Mean Error Differences for Question Groups

	Section Q	**Section A**
G1	0.01 (σ: 3.9)	0.07 (σ: 2.29)
G2	5.69 (σ: 13.44)	5.74 (σ: 16.13)
G3	5.01 (σ: 12.43)	6.15 (σ: 12.36)

VI. Discussion

The results showed that not many students demonstrate a reduction in semantic errors, regardless of the section they belong to. This finding indicates the difficulty of knowledge transfer in semantic aspects, consistent with previous studies that show the difficulty of semantic knowledge transfer between languages [37], [38]. Despite the fact that these problems involve the same abstract operations from an expert's perspective, students may struggle to recognize the conceptual similarities and effectively transfer their knowledge. To improve students' performance and enable smoother knowledge transfer between languages, it may be necessary to develop strategies that help trigger the knowledge connections between languages.

We also observed that students in both sections had higher semantic error rates in MongoDB compared to the other two languages. This suggests that students may face challenges in understanding the query language specific to document-based databases. Interestingly, we found that learning MongoDB after SQL (Section Q) resulted in better performance in MongoDB in terms of semantic errors, compared to those who learned MongoDB first (Section A). However, it seems that starting with MongoDB first did not notably assist students in reducing their semantic errors later in SQL. These findings suggest that learning SQL first may facilitate a better understanding of certain abstract data operation concepts. However, it is important to note that further research, particularly qualitative research involving student interviews, is needed to explore this matter.

Although a higher percentage of students in Section A demonstrated a reduction in semantic errors through their learning path compared to Section Q, this does not necessarily imply the superiority of MongoDB as the first language to be taught. It is important to consider multiple influential factors that may have contributed to this observation. One possible explanation is that students in Section A were initially exposed to a relatively error-prone language, leading to smoother progress as illustrated in Figure 3. Further research is needed to comprehensively understand the factors influencing students' performance and the effectiveness of different language instruction sequences.

In addition, the majority of students in both sections demonstrated a reduction in syntax errors among the studied question groups. This result aligns with the conclusion that practice helps students overcome syntax issues for CS1 students in procedural programming languages [13]. One potential explanation is that despite differences in question descriptions and goals, similar concept questions share a common high-level syntax structure. As such, once students solve a question, the structure and content of the previous solution may scaffold them in solving subsequent questions, leading to a reduction in syntax errors. Additionally, with practice and self-correction based on the course materials and online resources, students are building a mental model for that specific language which allows them to more effectively apply the language's syntax rules and avoid common errors. This finding suggests a potential solution to overcome syntax errors, which are often considered a stumbling block in the learning process [7]. It is noteworthy that although the majority of students demonstrated a reduction in syntax errors among those question groups, approximately 30% of the students did not show a decrease in syntax errors. Currently, we do not understand the reasons behind why some of the students were unable to

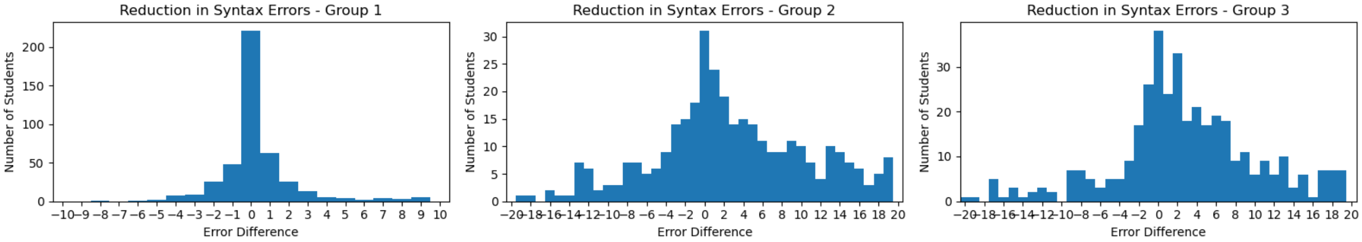

Fig. 4: Histograms showing the distribution of the reduction in syntax errors in three different question groups. The x-axis represents the Error Difference (defined in Section IV), with positive values indicating a decrease in errors. The y-axis indicates the number of students in each group that achieved a particular Error Difference.

reduce syntactic errors across all question groups. Therefore, a more detailed investigation is required to gain insights into the factors contributing to this phenomenon.

It was observed that students with fewer total submissions displayed a better ability to reduce errors on the initial or easier questions compared to students with a higher number of submissions. However, as the students progressed to more challenging or later questions, both groups showed a reduction in syntactic errors. This may indicate that students are capable of transferring syntactic knowledge, but the rate at which knowledge transfer occurs may not be the same across students. To better support high-submission students, it may be necessary to provide them with additional time for learning and allow sufficient time for knowledge transfer to occur.

Therefore, our research findings suggest that larger proportions of students were able to successfully transfer syntactic knowledge in the context of the same query language, since it may be considered near transfer. However, it is important for instructors to recognize that students may not readily apply abstract data operations across different languages, even if the operations seem similar from the instructor's perspective. This is because achieving far transfer in such cases is inherently more challenging.

VII. Limitations and Future Work

Our study relied on the comparison of students' homework errors as a measurement of their performance and knowledge transfer. We acknowledge that this approach may not always provide an accurate assessment. One of the reasons is that there are no restrictions on the number of submission attempts, and students have a two-week period to work on the assignments. This allows them multiple attempts and applications of the trial and error mentality, leading to fluctuations in the error counts. Additionally, while we attempted to select questions of similar difficulty and potential answer length, we cannot guarantee absolute uniformity in difficulty across all questions. This variability in question difficulty may have contributed to fluctuations in the number of submissions from students. To further enhance the quantitative research in this area, a potential avenue could involve a more detailed classification of assignments based on their concepts, length, difficulty levels, etc. and a categorization of students based on their distinct learning methods. This approach would enable a more comprehensive exploration of how various learning methods impact knowledge transfer and performance while minimizing the influence of confounding factors.

Furthermore, due to the lack of categorization of semantic errors, we do not know the exact abstract data operations corresponding to each semantic error students made. So, our current study does not know which abstract operation is causing the fluctuations in the number of errors that students made in complex problems that contain multiple required operations. Future research should prioritize the classification of semantic errors to gain a deeper understanding of students' errors and facilitate the provision of more targeted assistance.

Lastly, our analysis was conducted using data from a single university for a semester. Therefore, the generalizability of our findings to a larger scale is still uncertain. To further strengthen the validity of our conclusions, we plan to conduct future studies in different semesters and expand our analysis including both semantic and syntax aspects. We also encourage other researchers to replicate our methodology with data from multiple institutions to provide more insights into the findings presented in this study.

VIII. Conclusion

In this study, we gain an understanding of the transferability of query language knowledge and skills in the context of database education. Our investigation sought to uncover the potential for abstract data operation concept transfer across different database languages, and the effect of repeated practice in reducing syntax errors within the same query language.

Based on our findings, it appears that the teaching order of different database languages has an influence on the transfer of abstract data operations knowledge. However, regardless of the specific teaching order used, only a small percentage of students demonstrated a reduction in semantic errors which reflects the difficulty of semantic knowledge transfer between database query languages. On the contrary, in terms of syntax errors, the majority of students demonstrated a reduction in errors after practice in the same query language which suggests instructors can leverage focused practice within a specific language to address and overcome potential barriers in the learning process.

Acknowledgment

We would like to express our gratitude to Weihao Cheng for providing valuable input on refining the paper content and making it more accessible to non-CS majors. A special thanks goes to Ang Li for proposing enhancements to the code structure and providing suggestions on data visualizations.

References

[1] H. Lu, H. C. Chan, and K. K. Wei, "A survey on usage of sql," *SIGMOD Rec.*, vol. 22, no. 4, p. 60–65, dec 1993. [Online]. Available: https://doi.org/10.1145/166635.166656

[2] M. Guo, K. Qian, and L. Yang, "Hands-on labs for learning mobile and nosql database security," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, 2016, pp. 606–607.

[3] L. Li, K. Qian, Q. Chen, R. Hasan, and G. Shao, "Developing hands-on labware for emerging database security," in *Proceedings of the 17th Annual Conference on Information Technology Education*, ser. SIGITE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 60–64. [Online]. Available: https://doi.org/10.1145/2978192.2978225

[4] S. Mohan, "Teaching nosql databases to undergraduate students: A novel approach," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 314–319. [Online]. Available: https://doi.org/10.1145/3159450.3159554

[5] B. Fowler, J. Godin, and M. Geddy, "Teaching case: introduction to nosql in a traditional database course," *Journal of Information Systems Education*, vol. 27, no. 2, p. 99, 2016.

[6] A. Alawini, P. Rao, L. Zhou, L. Kang, and P.-C. HO, "Work-in-progress: Triql: A tool for learning relational, graph and document-oriented database programming." in *2021 Illinois-Indiana Regional Conference*, 2021.

[7] S. Poulsen, L. Butler, A. Alawini, and G. L. Herman, "Insights from student solutions to sql homework problems," in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 2020, pp. 404–410.

[8] A. Ahadi, V. Behbood, A. Vihavainen, J. Prior, and R. Lister, "Students' semantic mistakes in writing seven different types of sql queries," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 272–277. [Online]. Available: https://doi.org/10.1145/2899415.2899464

[9] ——, "Students' syntactic mistakes in writing seven different types of sql queries and its application to predicting students' success," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, USA: ACM, 2016, pp. 401–406.

[10] T. Taipalus, M. Siponen, and T. Vartiainen, "Errors and complications in sql query formulation," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 3, pp. 1–29, 2018.

[11] T. Taipalus and P. Perälä, "What to expect and what to focus on in sql query teaching," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 198–203. [Online]. Available: https://doi.org/10.1145/3287324.3287359

[12] A. Ahadi, J. Prior, V. Behbood, and R. Lister, "A quantitative study of the relative difficulty for novices of writing seven different types of sql queries," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15. New York, NY, USA: ACM, 2015, p. 201–206.

[13] J. Edwards, J. Ditton, D. Trninic, H. Swanson, S. Sullivan, and C. Mano, "Syntax exercises in cs1," in *Proceedings of the 2020 ACM Conference on International Computing Education Research*, ser. ICER '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 216–226. [Online]. Available: https://doi.org/10.1145/3372782.3406259

[14] S. Overflow, "Stack overflow developer survey 2019," 2019.

[15] A. Sellers, "10 top programming languages to learn in 2023 (in-demand)," feb 2023. [Online]. Available: https://www.codingdojo.com/blog/top-programming-languages

[16] "Sql – a universal language for working with databases," jun 2020. [Online]. Available: https://www.sqlsplus.com/sql-a-universal-language-for-working-with-databases/

[17] D. Miedema, E. Aivaloglou, and G. Fletcher, "Identifying sql misconceptions of novices: Findings from a think-aloud study," in *Proceedings of the 17th ACM Conference on International Computing Education Research*, ser. ICER 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 355–367. [Online]. Available: https://doi.org/10.1145/3446871.3469759

[18] D. Miedema, G. Fletcher, and E. Aivaloglou, "So many brackets! an analysis of how sql learners (mis)manage complexity during query formulation," in *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, ser. ICPC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 122–132. [Online]. Available: https://doi.org/10.1145/3524610.3529158

[19] A. Migler and A. Dekhtyar, "Mapping the sql learning process in introductory database courses," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 619–625. [Online]. Available: https://doi.org/10.1145/3328778.3366869

[20] S. Brass and C. Goldberg, "Semantic errors in sql queries: A quite complete list," *Journal of Systems and Software*, vol. 79, no. 5, pp. 630–644, 2006, quality Software. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016412120500124X

[21] D. Miedema and G. Fletcher, "Sqlvis: Visual query representations for supporting sql learners," in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2021, pp. 1–9.

[22] S. Yang, Z. Wei, G. L. Herman, and A. Alawini, "Analyzing patterns in student sql solutions via levenshtein edit distance," in *Proceedings of the Eighth ACM Conference on Learning @ Scale*, ser. L@S '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 323–326. [Online]. Available: https://doi.org/10.1145/3430895.3460979

[23] S. Yang, G. L. Herman, and A. Alawini, "Analyzing student sql solutions via hierarchical clustering and sequence alignment scores," in *1st International Workshop on Data Systems Education*, ser. DataEd '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 10–15. [Online]. Available: https://doi.org/10.1145/3531072.3535319

[24] ——, "Mining sql problem solving patterns using advanced sequence processing algorithms," in *Proceedings of the 2nd International Workshop on Data Systems Education: Bridging Education Practice with Education Research*, ser. DataEd '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 37–43. [Online]. Available: https://doi.org/10.1145/3596673.3596973

[25] "What is a non-relational database?" 2023. [Online]. Available: https://www.mongodb.com/databases/non-relational

[26] H. Kathuria, "The most popular databases for 2022," jan 2022. [Online]. Available: https://learnsql.com/blog/most-popular-databases-2022/

[27] D. Kotsifakos, D. Magetos, A. Veletsos, and C. Douligeris, "Teaching the basic commands of nosql databases using neo4j in vocational education and training (vet)," *European Journal of Engineering and Technology Research*, no. CIE, p. 13–18, Apr. 2019. [Online]. Available: https://www.ej-eng.org/index.php/ejeng/article/view/1291

[28] R. Alkhabaz, S. Poulsen, M. Chen, and A. Alawini, "Insights from student solutions to mongodb homework problems," in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 276–282. [Online]. Available: https://doi.org/10.1145/3430665.3456308

[29] J. Guia, V. G. Soares, and J. Bernardino, "Graph databases: Neo4j analysis." in *ICEIS (1)*, 2017, pp. 351–356.

[30] D. Fernandes and J. Bernardino, "Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb." in *Data*, 2018, pp. 373–380.

[31] M. Chen, S. Poulsen, R. Alkhabaz, and A. Alawini, "A quantitative analysis of student solutions to graph database problems," in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 283–289. [Online]. Available: https://doi.org/10.1145/3430665.3456314

[32] M. L. GICK and K. J. HOLYOAK, "Chapter 2 - the cognitive basis of knowledge transfer," in *Transfer of Learning*, S. M. Cormier and J. D. Hagman, Eds. San Diego: Academic Press, 1987, pp. 9–46. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780121889500500084

[33] L. Argote, P. Ingram, J. M. Levine, and R. L. Moreland, "Knowledge transfer in organizations: Learning from the experience of others," *Organizational Behavior and Human Decision*

Processes, vol. 82, no. 1, pp. 1–8, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0749597800928838

[34] S. M. Barnett and S. J. Ceci, “When and where do we apply what we learn?: A taxonomy for far transfer.” *Psychological bulletin*, vol. 128, no. 4, p. 612, 2002.

[35] E. F. Codd, “Data models in database management,” *SIGPLAN Not.*, vol. 16, no. 1, p. 112–114, jun 1980. [Online]. Available: https://doi.org/10.1145/960124.806891

[36] M. West, G. L. Herman, and C. B. Zilles, “Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning,” 2015.

[37] E. Tshukudu and Q. Cutts, “Semantic transfer in programming languages: Exploratory study of relative novices,” in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 307–313. [Online]. Available: https://doi.org/10.1145/3341525.3387406

[38] N. Shrestha, C. Botta, T. Barik, and C. Parnin, “Here we go again: Why is it difficult for developers to learn another programming language?” in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 691–701. [Online]. Available: https://doi.org/10.1145/3377811.3380352